Configuring Media-Independent PPP and Multilink PPP

This chapter describes how to configure the PPP and Multilink PPP (MLP) features that can be configured on any interface. It includes the following main sections:

- PPP Encapsulation Overview
- PPP Configuration Task List
- Configuring MLP Interleaving and Queueing for Real-Time Traffic
- MLP Inverse Multiplexer and Distributed MLP Configuration Task List
- Monitoring and Maintaining PPP and MLP Interfaces
- PPP and MLP Configuration Examples

This chapter also describes address pooling for point-to-point links, which is available on all asynchronous serial, synchronous serial, and ISDN interfaces. See the chapter “Configuring Asynchronous SLIP and PPP” in this publication for information about PPP features and requirements that apply only to asynchronous lines and interfaces.

For a complete description of the PPP commands in this chapter, see the *Cisco IOS Dial Services Command Reference* publication. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

PPP Encapsulation Overview

PPP, described in RFC 1661, encapsulates network layer protocol information over point-to-point links. You can configure PPP on the following types of physical interfaces:

- Asynchronous serial
- HSSI
- ISDN
- Synchronous serial

By enabling PPP encapsulation on physical interfaces, PPP can also be in effect on calls placed by the dialer interfaces that use the physical interfaces.

The current implementation of PPP supports option 3, authentication using Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP), option 4, Link Quality Monitoring (LQM), and option 5, Magic Number configuration options. The software always sends option 5 and negotiates for options 3 and 4 if so configured. All other options are rejected.

Magic Number support is available on all serial interfaces. PPP always attempts to negotiate for Magic Numbers, which are used to detect looped-back lines. Depending on how the **down-when-looped** command is configured, the router might shut down a link if it detects a loop.

The software provides the CHAP and PAP on serial interfaces running PPP encapsulation. For detailed information about authentication, see the *Cisco IOS Security Configuration Guide*.

With Cisco IOS Release 11.2 F, Cisco now supports fast switching of incoming and outgoing DECnet and CLNS packets over PPP.

PPP Configuration Task List

To configure PPP on a serial interface (including ISDN), perform the following required task in interface configuration mode: Enabling PPP Encapsulation (required for PPP encapsulation).

You can also complete the tasks in the following sections; these tasks are optional but offer a variety of uses and enhancements for PPP on your systems and networks:

- Enabling CHAP or PAP Authentication
- Enabling Link Quality Monitoring
- Configuring Compression of PPP Data
- Configuring Microsoft Point-to-Point Compression
- Configuring IP Address Pooling
- Configuring PPP Reliable Link
- Disabling or Reenabling Peer Neighbor Routes
- Configuring PPP Half-Bridging
- Configuring Multilink PPP
- Configuring MLP Interleaving and Queueing for Real-Time Traffic
- MLP Inverse Multiplexer and Distributed MLP Configuration Task List

See the section “Monitoring and Maintaining PPP and MLP Interfaces” later in this chapter for tips on maintaining PPP. See the “PPP and MLP Configuration Examples” at the end of this chapter for ideas on how to implement PPP and MLP in your network.

Enabling PPP Encapsulation

To enable PPP on serial lines to encapsulate IP and other network protocol datagrams, use the following command in interface configuration mode:

Command	Purpose
`encapsulation ppp`	Enables PPP encapsulation.

PPP echo requests are used as keepalives to minimize disruptions to the end users of your network. The **no keepalive command** can be used to disable echo requests.

Enabling CHAP or PAP Authentication

PPP with CHAP or PAP authentication is often used to inform the central site about which remote routers are connected to it.

With this authentication information, if the router or access server receives another packet for a destination to which it is already connected, it does not place an additional call. However, if the router or access server is using rotaries, it sends the packet out the correct port.

CHAP and PAP were originally specified in RFC 1334, and CHAP is updated in RFC 1994. These protocols are supported on synchronous and asynchronous serial interfaces. When using CHAP or PAP authentication, each router or access server identifies itself by a *name*. This identification process prevents a router from placing another call to a router to which it is already connected, and also prevents unauthorized access.

Access control using CHAP or PAP is available on all serial interfaces that use PPP encapsulation. The authentication feature reduces the risk of security violations on your router or access server. You can configure either CHAP or PAP for the interface.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the local router or access server sends a CHAP packet to the remote device. The CHAP packet requests or “challenges” the remote device to respond. The challenge packet consists of an ID, a random number, and the host name of the local router.

The required response has two parts:

- An encrypted version of the ID, a secret password, and the random number
- Either the host name of the remote device or the name of the user on the remote device

When the local router or access server receives the response, it verifies the secret password by performing the same encryption operation as indicated in the response and looking up the required host name or username. The secret passwords must be identical on the remote device and the local router.

Because this response is sent, the password is never sent in clear text, preventing other devices from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local router.

CHAP transactions occur only when a link is established. The local router or access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the local router or access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the local router or access server requires authentication from remote devices. If the remote device does not support the enabled protocol, no traffic will be passed to that device.

To use CHAP or PAP, you must perform the following tasks:

- Enable PPP encapsulation.
- Enable CHAP or PAP on the interface.
- For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

To enable PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
encapsulation ppp	Enables PPP encapsulation on an interface.

To enable CHAP or PAP authentication on an interface configured for PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
ppp authentication {**chap** | **chap pap** | **pap chap** | **pap**} [**if-needed**] [*list-name* | **default**] [**callin**]	Defines the authentication methods supported and the order in which they are used.

The **ppp authentication chap** optional keyword **if-needed** can be used only with Terminal Access Controller Access Control System (TACACS) or extended TACACS.

With authentication, authorization, and accounting (AAA) configured on the router and list names defined for AAA, the *list-name* optional keyword can be used with AAA/TACACS+.

Caution If you use a *list-name* that has not been configured with the **aaa authentication ppp** command, you disable PPP on the line.

Add a **username** entry for each remote system from which the local router or access server requires authentication.

To specify the password to be used in CHAP or PAP caller identification, use the following command in global configuration mode:

Command	Purpose
username *name* [**user-maxlinks** *link-number*] **password** *secret*	Configures identification. Optionally, you can specify the maximum number of connections a user can establish. To use the **user-maxlinks** keyword, you must also use the **aaa authorization network default local** command, and PPP encapsulation and name authentication on all the interfaces the user will be accessing.

Make sure this password does not include spaces or underscores.

To configure TACACS on a specific interface as an alternative to global host authentication, use one of the following command in interface configuration mode:

Command	Purpose
ppp use-tacacs [**single-line**] or **aaa authentication ppp**	Configures TACACS.

Use the **ppp use-tacacs** command with TACACS and Extended TACACS. Use the **aaa authentication ppp** command with AAA/TACACS+.

For an example of CHAP, see the section "CHAP with an Encrypted Password" at the end of this chapter. CHAP is specified in RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*.

Enabling Link Quality Monitoring

Link Quality Monitoring (LQM) is available on all serial interfaces running PPP. LQM will monitor the link quality, and if the quality drops below a configured percentage, the router will shut down the link. The percentages are calculated for both the incoming and outgoing directions. The outgoing quality is calculated by comparing the total number of packets and bytes sent with the total number of packets and bytes received by the destination node. The incoming quality is calculated by comparing the total number of packets and bytes received with the total number of packets and bytes sent by the destination peer.

When LQM is enabled, Link Quality Reports (LQRs) are sent, in place of keepalives, every keepalive period. All incoming keepalives are responded to properly. If LQM is not configured, keepalives are sent every keepalive period and all incoming LQRs are responded to with an LQR.

LQR is specified in RFC 1989, *PPP Link Quality Monitoring*, by William A. Simpson of Computer Systems Consulting Services.

To enable LQM on the interface, use the following command in interface configuration mode:

Command	Purpose
ppp quality *percentage*	Enables LQM on the interface.

The *percentage* argument specifies the link quality threshold. That percentage must be maintained, or the link is deemed to be of poor quality and taken down.

Configuring Compression of PPP Data

You can configure point-to-point software compression on serial interfaces that use PPP encapsulation. Compression reduces the size of a PPP frame via lossless data compression. PPP encapsulations support both predictor and Stacker compression algorithms.

If most of your traffic is already compressed files, do not use compression.

Most routers support software compression only, but in the Cisco 7000 series routers hardware compression and distributed compression are also available, depending on the interface processor and compression service adapter hardware installed in the router.

To configure compression, complete the tasks in one of the following sections:

- Software Compression
- Hardware-Dependent Compression

Software Compression

Software compression is available in all router platforms. Software compression is performed by the main processor in the router.

Compression is performed in software and might significantly affect system performance. We recommend that you disable compression if the router CPU load exceeds 65 percent. To display the CPU load, use the **show process cpu** EXEC command.

To configure compression over PPP, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`encapsulation ppp`	Enables encapsulation of a single protocol on the serial line.
Step 2	`compress [predictor | stac | mppc [ignore-pfc]]`	Enables compression.

Hardware-Dependent Compression

When you configure Stacker compression on Cisco 7000 series routers with a 7000 Series Route Switch Processor (RSP7000), on Cisco 7200 series routers, and on Cisco 7500 series routers, there are three methods of compression: hardware compression, distributed compression, and software compression.

Hardware and distributed compression are available on routers that have the SA-Comp/1 and SA-Comp/4 data compression service adapters (CSAs). CSAs are available on Cisco 7200 series routers, on Cisco 7500 series routers with second-generation Versatile Interface Processors (VIP2s), and on Cisco 7000 series routers with the RSP7000 and 7000 Series Chassis Interface (RSP7000CI). (CSAs require VIP2 model VIP2-40.)

To configure hardware or distributed compression over PPP, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`encapsulation ppp`	Enables encapsulation of a single protocol on the serial line.
Step 2	`compress stac [distributed | software]` (Cisco 7000 series with RSP7000 and Cisco 7500 series) or `compress stac [csa slot | software]` (Cisco 7200 series)	Enables compression.

Specifying the **compress stac** command with no options causes the router to use the fastest available compression method:

- If the router contains a CSA, compression is performed in the CSA hardware (hardware compression).
- If the CSA is not available, compression is performed in the software installed on the VIP2 (distributed compression).
- If the VIP2 is not available, compression is performed in the main processor of the router (software compression).

Using hardware compression in the CSA frees the main processor of the router for other tasks. You can also configure the router to use the VIP2 to perform compression by using the **distributed** option, or to use the main processor of the router by using the **software** option. If the VIP2 is not available, compression is performed in the main processor of the router.

When compression is performed in software installed in the main processor of the router, it might substantially affect system performance. We recommend that you disable compression in the main processor of the router if the router CPU load exceeds 40 percent. To display the CPU load, use the **show process cpu** EXEC command.

Specifying the **compress stac** command with no options causes the router to use the fastest available compression method.

Configuring Microsoft Point-to-Point Compression

Microsoft Point-to-Point Compression (MPPC) is a scheme used to compress PPP packets between Cisco and Microsoft client devices. The MPPC algorithm is designed to optimize bandwidth utilization in order to support multiple simultaneous connections. The MPPC algorithm uses a Lempel-Ziv (LZ)-based algorithm with a continuous history buffer called a dictionary.

The Compression Control Protocol (CCP) configuration option for MPPC is 18.

Exactly one MPPC datagram is encapsulated in the PPP information field. The PPP protocol field indicates the hexadecimal type of 00FD for all compressed datagrams. The maximum length of the MPPC datagram sent over PPP is the same as the MTU of the PPP interface; however, this length cannot be greater than 8192 bytes because the history buffer is limited to 8192 bytes. If compressing the data results in data expansion, the original data is sent as an uncompressed MPPC packet.

The history buffers between compressor and decompressor are synchronized by maintaining a 12-bit coherency count. If the decompressor detects that the coherency count is out of sequence, then the error recovery process described next is performed:

1. Reset Request (RR) packet is sent from the decompressor.
2. The compressor then flushes the history buffer and sets the flushed bit in the next packet it sends.
3. Upon receiving the flushed bit set packet, the decompressor flushes the history buffer.

Synchronization is achieved without CCP using the Reset Acknowledge (RA) packet, which can consume additional time.

Compression negotiation between a router and a Windows 95 client occurs through the following process:

1. Windows 95 sends a request for both STAC (option 17) and MPPC (option 18) compression.
2. The router sends a negative acknowledgment (NAK) requesting only MPPC.
3. Windows 95 resends the request for MPPC.
4. The router sends an acknowledgment (ACK) confirming MPPC compression negotiation.

MPPC Restrictions

The following restrictions apply to the MPPC feature:

- MPPC is only supported with PPP encapsulation.
- Compression can be processor intensive because it requires a reserved block of memory to maintain the history buffer. Do not enable modem or hardware compression because it may cause performance degradation, compression failure, or data expansion.
- Both ends of the point-to-point link must be using the same compression method (STAC, Predictor, or MPPC, for example).

Configuring MPPC

PPP encapsulation must be enabled before you can configure MPPC. For information on how to configure PPP encapsulation see the section "Enabling PPP Encapsulation" earlier in this chapter.

There is only one command required to configure MPPC. The existing **compress** command supports the **mppc** keyword, which prepares the interface to initiate CCP and negotiates MPPC with the Microsoft client. To set MPPC once PPP encapsulation is configured on the router, use the following command in interface configuration mode:

Command	Purpose
compress [**mppc** [**ignore-pfc**]]	Enables MPPC on the interface.

The **ignore-pfc** keyword instructs the router to ignore the protocol field compression flag negotiated by LCP. For example, the uncompressed standard protocol field value for IP is 0x0021 and 0x21 when compression is enabled. When the **ignore-pfc** option is enabled, the router will continue to use the uncompressed value (0x0021). Using the **ignore-pfc** option is helpful for some asynchronous driver devices that use an uncompressed protocol field (0x0021), even though the protocol field compression is negotiated between peers. Figure 49 displays protocol rejections when the **debug ppp negotiation** command is enabled. These errors can be remedied by setting the **ignore-pfc** option.

Figure 49 Sample debug ppp negotiation Command Output Showing Protocol Reject

```
PPP Async2: protocol reject received for protocol = 0x2145
PPP Async2: protocol reject received for protocol = 0x2145
PPP Async2: protocol reject received for protocol = 0x2145
```

Configuring IP Address Pooling

A point-to-point interface must be able to provide a remote node with its IP address through the IP Control Protocol (IPCP) address negotiation process. The IP address can be obtained from a variety of sources. The address can be configured through the command line, entered with an EXEC-level command, provided by TACACS+ or the Dynamic Host Configuration Protocol (DHCP), or from a locally administered pool.

IP address pooling uses a pool of IP addresses from which an incoming interface can provide an IP address to a remote node through IPCP address negotiation process. IP address pooling also enhances configuration flexibility by allowing multiple types of pooling to be active simultaneously.

See the chapter "Configuring Asynchronous SLIP and PPP" in this publication for additional information about address pooling on asynchronous interfaces and about the Serial Line Internet Protocol (SLIP).

Peer Address Allocation

A peer IP address can be allocated to an interface through several methods:

- Dialer map lookup—This method is used only if the peer requests an IP address, no other peer IP address has been assigned, and the interface is a member of a dialer group.
- PPP or SLIP EXEC command—An asynchronous dialup user can enter a peer IP address or host name when PPP or SLIP is invoked from the command line. The address is used for the current session and then discarded.

- IPCP negotiation—If the peer presents a peer IP address during IPCP address negotiation and no other peer address is assigned, the presented address is acknowledged and used in the current session.
- Default IP address—The **peer default ip address** command and the **member peer default ip address** command can be used to define default peer IP addresses.
- TACACS+ assigned IP address—During the authorization phase of IPCP address negotiation, TACACS+ can return an IP address that the user being authenticated on a dialup interface can use. This address overrides any default IP address and prevents pooling from taking place.
- DHCP retrieved IP address—If configured, the routers acts as a proxy client for the dialup user and retrieves an IP address from a DHCP server. That address is returned to the DHCP server when the timer expires or when the interface goes down.
- Local address pool—The local address pool contains a set of contiguous IP addresses (a maximum of 1024 addresses) stored in two queues. The free queue contains addresses available to be assigned and the used queue contains addresses that are in use. Addresses are stored to the free queue in first-in, first-out (FIFO) order to minimize the chance the address will be reused, and to allow a peer to reconnect using the same address that it used in the last connection. If the address is available, it is assigned; if not, another address from the free queue is assigned.
- Chat script (asynchronous serial interfaces only)—The IP address in the **dialer map** command entry that started the script is assigned to the interface and overrides any previously assigned peer IP address.
- Virtual terminal/protocol translation—The translate command can define the peer IP address for a virtual terminal (pseudo asynchronous interface).
- The pool configured for the interface is used, unless TACACS+ returns a pool name as part of AAA. If no pool is associated with a given interface, the global pool named default is used.

Precedence Rules

The following precedence rules of peer IP address support determine which address is used. Precedence is listed from most likely to least likely:

1. AAA/TACACS+ provided address or addresses from the pool named by AAA/TACACS+
2. An address from a local IP address pool or DHCP (typically not allocated unless no other address exists)
3. Dialer map lookup address (not done unless no other address exists)
4. Address from an EXEC-level PPP or SLIP command, or from a chat script
5. Configured address from the **peer default ip address** command or address from the protocol **translate** command
6. Peer provided address from IPCP negotiation (not accepted unless no other address exists)

Interfaces Affected

Address pooling is available on all asynchronous serial, synchronous serial, ISDN BRI, and ISDN PRI interfaces running PPP.

Choosing the IP Address Assignment Method

The IP address pooling feature now allows configuration of a global default address pooling mechanism, per-interface configuration of the address pooling mechanism, and per-interface configuration of a specific address or pool name.

You can define the type of IP address pooling mechanism used on router interfaces in one or both of the ways described in the following sections:

- Defining the Global Default Address Pooling Mechanism
- Configuring IP Address Assignment

Defining the Global Default Address Pooling Mechanism

The global default mechanism applies to all point-to-point interfaces that support PPP encapsulation and that have not otherwise been configured for IP address pooling. You can define the global default mechanism to be either DHCP or local address pooling.

To configure the global default mechanism for IP address pooling, perform the tasks in one of following sections:

- Defining DHCP as the Global Default Mechanism
- Defining Local Address Pooling as the Global Default Mechanism

After you have defined a global default mechanism, you can disable it on a specific interface by configuring the interface for some other pooling mechanism. You can define a local pool other than the default pool for the interface or you can configure the interface with a specific IP address to be used for dial-in peers.

Defining DHCP as the Global Default Mechanism

DHCP specifies the following components:

- A DHCP server—A host-based DHCP server configured to accept and process requests for temporary IP addresses.
- A DHCP proxy-client—A Cisco access server configured to arbitrate DHCP calls between the DHCP server and the DHCP client. The DHCP client-proxy feature manages a pool of IP addresses available to dial-in clients without a known IP address.

To enable DHCP as the global default mechanism, use the following commands in global configuration mode:

	Command	Purpose
Step 1	**ip address-pool dhcp-proxy-client**	Specifies DHCP client-proxy as the global default mechanism.
Step 2	**ip dhcp-server** [*ip-address* | *name*]	(Optional) Specifies the IP address of a DHCP server for the proxy client to use.

In Step 2, you can provide as few as one or as many as ten DHCP servers for the proxy-client (the Cisco router or access server) to use. DHCP servers provide temporary IP addresses.

Defining Local Address Pooling as the Global Default Mechanism

To specify that the global default mechanism to use is local pooling, use the following commands in global configuration mode:

	Command	Purpose
Step 1	**ip address-pool local**	Specifies local pooling as the global default mechanism.
Step 2	**ip local pool** {**default** | *pool-name*} *low-ip-address* [*high-ip-address*]	Creates one or more local IP address pools.

If no other pool is defined, the local pool called default is used.

Configuring IP Address Assignment

After you have defined a global default mechanism for assigning IP addresses to dial-in peers, you can then configure the few interfaces for which it is important to have a nondefault configuration. You can do any of the following;

- Define a nondefault address pool for use by a specific interface.
- Define DHCP on an interface even if you have defined local pooling as the global default mechanism.
- Specify one IP address to be assigned to all dial-in peers on an interface.
- Make temporary IP addresses available on a per-interface basis to asynchronous clients using SLIP or PPP.

To define a nondefault address pool for use on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**ip local pool** *poolname* {*low-ip-address* [*high-ip-address*]}	Creates one or more local IP address pools.
Step 2	**interface** *type number*	Specifies the interface and enters interface configuration mode.
Step 3	**peer default ip address pool** *pool-name*	Specifies the pool for the interface to use.

To define DHCP as the IP address mechanism for an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**interface** *type number*	Specifies the interface and enters interface configuration mode.
Step 2	**peer default ip address pool dhcp**	Specifies DHCP as the IP address mechanism on this interface.

To define a specific IP address to be assigned to all dial-in peers on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**interface** *type number*	Specifies the interface and enters interface configuration mode.
Step 2	**peer default ip address** *ip-address*	Specifies the IP address to assign.

Configuring PPP Reliable Link

PPP reliable link is Cisco's implementation of RFC 1663, *PPP Reliable Transmission*, which defines a method of negotiating and using Numbered Mode Link Access Procedure, Balanced (LAPB) to provide a reliable serial link. Numbered Mode LAPB provides retransmission of error packets across the serial link.

Although LAPB protocol overhead consumes some bandwidth, you can offset that consumption by the use of PPP compression over the reliable link. PPP compression is separately configurable and is not required for use of a reliable link.

Note PPP reliable link is available only on synchronous serial interfaces, including ISDN BRI and ISDN PRI interfaces. PPP reliable link cannot be used over V.120, and does not work with Multilink PPP.

To configure PPP reliable link on a specified interface, use the following command in interface configuration mode:

Command	Purpose
ppp reliable-link	Enables PPP reliable link.

Having reliable links enabled does not guarantee that all connections through the specified interface will in fact use reliable link. It only guarantees that the router will attempt to negotiate reliable link on this interface.

Troubleshooting PPP

You can troubleshoot PPP reliable link by using the **debug lapb** command and the **debug ppp negotiations**, **debug ppp errors**, and **debug ppp packets** commands. You can determine whether LAPB has been established on a connection by using the **show interface** command.

Disabling or Reenabling Peer Neighbor Routes

The Cisco IOS software automatically creates neighbor routes by default; that is, it automatically sets up a route to the peer address on a point-to-point interface when the PPP IPCP negotiation is completed.

To disable this default behavior, or to reenable it once it has been disabled, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`no peer neighbor-route`	Disables creation of neighbor routes.
Step 2	`peer neighbor-route`	Reenables creation of neighbor routes.

Note If entered on a dialer or asynchronous group interface, this command affects all member interfaces.

Configuring PPP Half-Bridging

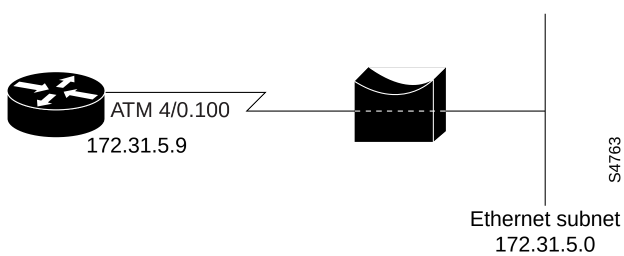
For situations in which a routed network needs connectivity to a remote bridged Ethernet network, a serial or ISDN interface can be configured to function as a PPP half-bridge. The line to the remote bridge functions as a virtual Ethernet interface, and the serial or ISDN interface on the router functions as a node on the same Ethernet subnetwork as the remote network.

The bridge sends bridge packets to the PPP half-bridge, which converts them to routed packets and forwards them to other router processes. Likewise, the PPP half-bridge converts routed packets to Ethernet bridge packets and sends them to the bridge on the same Ethernet subnetwork.

Note An interface cannot function as both a half-bridge and a bridge.

Figure 50 shows a router with a serial interface configured as a PPP half-bridge. The interface functions as a node on the Ethernet subnetwork with the bridge. Note that the serial interface has an IP address on the same Ethernet subnetwork as the bridge.

Figure 50 Router Serial Interface Configured as a Half-Bridge



Note The Cisco IOS software supports no more than one PPP half-bridge per Ethernet subnetwork.

To configure a serial interface to function as a half-bridge, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**interface serial** *number*	Specifies the interface (and enter interface configuration mode).
Step 2	**ppp bridge appletalk** **ppp bridge ip** **ppp bridge ipx** [**novell-ether** | **arpa** | **sap** | **snap**]	Enables PPP half-bridging for one or more routed protocols: AppleTalk, IP, or Internet Protocol Exchange (IPX).
Step 3	**ip address** *n.n.n.n* **appletalk address** *network.node* **appletalk cable-range** *cable-range network.node* **ipx network** *network*	Provides a protocol address on the same subnetwork as the remote network.

Note You must enter the **ppp bridge** command either when the interface is shut down or before you provide a protocol address for the interface.

For more information about AppleTalk addressing, see the "Configuring AppleTalk" chapter; for more information about IPX addresses and encapsulations, see the "Configuring Novell IPX" chapter. Both chapters are in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

Configuring Multilink PPP

The Multilink PPP feature provides load balancing functionality over multiple WAN links, while providing multivendor interoperability, packet fragmentation and proper sequencing, and load calculation on both inbound and outbound traffic. The Cisco implementation of MLP supports the fragmentation and packet sequencing specifications in RFC 1717.

MLP allows packets to be fragmented and the fragments to be sent at the same time over multiple point-to-point links to the same remote address. The multiple links come up in response to a defined dialer load threshold. The load can be calculated on inbound traffic, outbound traffic, or on either, as needed for the traffic between the specific sites. MLP provides bandwidth on demand and reduces transmission latency across WAN links.

MLP is designed to work over the following types of single or multiple interfaces that have been configured to support both dial-on-demand rotary groups and PPP encapsulation:

- Synchronous serial interfaces
- Asynchronous serial interfaces
- BRI interfaces
- PRI interfaces

Configuring MLP on Synchronous Interfaces

To configure Multilink PPP on synchronous interfaces, you configure the synchronous interfaces to support PPP encapsulation and Multilink PPP.

To configure a synchronous interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**interface serial** *number*	Specifies an asynchronous interface.
Step 2	**no ip address**	Specifies no IP address for the interface.
Step 3	**encapsulation ppp**	Enables PPP encapsulation.
Step 4	**no fair-queue**	Disables WFQ on the interface.
Step 5	**ppp multilink**	Enables Multilink PPP.
Step 6	**pulse-time** *seconds*	Enables pulsing DTR signal intervals on the interface.

Repeat these steps for additional synchronous interfaces, as needed.

Configuring MLP on Asynchronous Interfaces

To configure MLP on asynchronous interfaces, configure the asynchronous interfaces to support dial-on-demand routing (DDR) and PPP encapsulation, then configure a dialer interface to support PPP encapsulation, bandwidth on demand, and Multilink PPP.

To configure an asynchronous interface to support DDR and PPP encapsulation, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**interface async** *number*	Specifies an asynchronous interface.
Step 2	**no ip address**	Specifies no IP address for the interface.
Step 3	**encapsulation ppp**	Enables PPP encapsulation.
Step 4	**dialer in-band**	Enables DDR on the interface.
Step 5	**dialer rotary-group** *number*	Includes the interface in a specific dialer rotary group.

Repeat these steps for additional asynchronous interfaces, as needed.

At some point, adding more asynchronous interfaces does not improve performance, With the default maximum transmission unit (MTU) size, MLP should support three asynchronous interfaces using V.34 modems. However, packets might be dropped occasionally if the maximum transmission unit (MTU) size is small or large bursts of short frames occur.

To configure a dialer interface to support PPP encapsulation and Multilink PPP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**interface dialer** *number*	Defines a dialer rotary group.
Step 2	**no ip address**	Specifies no IP address for the interface.
Step 3	**encapsulation ppp**	Enables PPP encapsulation.
Step 4	**dialer in-band**	Enables DDR on the interface.

	Command	Purpose
Step 5	**dialer load-threshold** *load* [**inbound** | **outbound** | **either**]	Configures bandwidth on demand by specifying the maximum load before the dialer places another call to a destination.
Step 6	**ppp multilink**	Enables Multilink PPP.

Configuring MLP on a Single ISDN BRI Interface

To enable MLP on a single ISDN BRI interface, you are not required to define a dialer rotary group separately because ISDN interfaces are dialer rotary groups by default.

To enable PPP on an ISDN BRI interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**interface bri** *number*	Specifies an interface.
Step 2	**ip address** *ip-address mask* [**secondary**]	Provides an appropriate protocol address for the interface.
Step 3	**encapsulation ppp**	Enables PPP encapsulation.
Step 4	**dialer idle-timeout** *seconds*	(Optional) Specifies a dialer idle timeout.
Step 5	**dialer load-threshold** *load*	Specifies the dialer load threshold for bringing up additional WAN links.
Step 6	**dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]	Configures the ISDN interface to call the remote site.
Step 7	**dialer-group** *group-number*	Controls access to this interface by adding it to a dialer access group.
Step 8	**ppp authentication pap**	(Optional) Enables PPP authentication.
Step 9	**ppp multilink**	Enables MLP on the dialer rotary group.

If you do not use PPP authentication procedures (Step 8), your telephone service must pass caller ID information.

The load threshold number is required. For an example of configuring MLP on a single ISDN BRI interface, see the section "MLP on One ISDN BRI Interface" at the end of this chapter.

When MLP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a very high idle timer. (The **dialer-load threshold 1** command no longer keeps a multilink bundle of *n* links connected indefinitely, and the **dialer-load threshold 2** command no longer keeps a multilink bundle of two links connected indefinitely.)

When you use dialer profiles with Multilink PPP, you must specify the **ppp multilink** command on both the physical and logical interfaces in order for Multilink PPP to be negotiated by the router. See the "Multilink PPP with Dialer Profiles Example" for more information.

Configuring MLP on Multiple ISDN BRI Interfaces

To enable MLP on multiple ISDN BRI interfaces, set up a dialer rotary interface and configure it for Multilink PPP, then configure the BRI interfaces separately and add them to the same rotary group.

To set up the dialer rotary interface for the BRI interfaces, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**interface dialer** *number*	Specifies the dialer rotary interface.
Step 2	**ip address** *address mask*	Specifies the protocol address for the dialer rotary interface.
Step 3	**encapsulation ppp**	Enables PPP encapsulation.
Step 4	**dialer in-band**	Specifies in-band dialing.
Step 5	**dialer idle-timeout** *seconds*	(Optional) Specifies the dialer idle timeout period, using the same timeout period as the individual BRI interfaces.
Step 6	**dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[:*isdn-subaddress*]]	Maps the next hop protocol address and name to the dial string needed to reach it.
Step 7	**dialer load-threshold** *load*	Specifies the dialer load threshold, using the same threshold as the individual BRI interfaces.
Step 8	**dialer-group** *number*	Controls access to this interface by adding it to a dialer access group.
Step 9	**ppp authentication chap**	(Optional) Enables PPP CHAP authentication.
Step 10	**ppp multilink**	Enables Multilink PPP.

If you do not use PPP authentication procedures (Step 10), your telephone service must pass caller ID information.

To configure each of the BRI interfaces to belong to the same rotary group, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**interface bri** *number*	Specifies one of the BRI interfaces.
Step 2	**no ip address**	Specifies that it does not have an individual protocol address.
Step 3	**encapsulation ppp**	Enables PPP encapsulation.
Step 4	**dialer idle-timeout** *seconds*	Sets the dialer idle timeout period, using the same timeout for each of the BRI interfaces you configure.
Step 5	**dialer rotary-group** *number*	Adds the interface to the rotary group.
Step 6	**dialer load-threshold** *load*	Specifies the dialer load threshold for bringing up additional WAN links.

Repeat Steps 1 through 6 for each BRI you want to belong to the same dialer rotary group.

When MLP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a very high idle timer. (The **dialer load-threshold 1** command no longer keeps a multilink bundle of *n* links connected indefinitely and the **dialer load-threshold 2** command no longer keeps a multilink bundle of two links connected indefinitely.)

Note Previously, when MLP was used in a dialer profile, a virtual access interface was always created as the bundle. It was bound to both the B channel and the dialer profile interfaces after creation and cloning. The dialer profile interface could act as the bundle without help from a virtual access interface. But with the Dynamic Multiple Encapsulations feature available in in Cisco IOS Release 12.1, it is no longer the virtual access interface that is added into the connected group of the dialer profile, but the dialer profile itself. The dialer profile becomes a connected member of its own connected group. See the "Dynamic Multiple Encapsulations over ISDN" example in the chapter "Configuring Peer-to-Peer DDR with Dialer Profiles" in this publication, for more information about dynamic multiple encapsulations and its relation to Multilink PPP.

For an example of configuring MLP on multiple ISDN BRI interfaces, see the "MLP on Multiple ISDN IBRI Interfaces" example at the end of this chapter.

Configuring MLP Interleaving and Queueing for Real-Time Traffic

Interleaving on MLP allows large packets to be multilink encapsulated and fragmented into a small enough size to satisfy the delay requirements of real-time traffic; small real-time packets are not multilink encapsulated and are sent between fragments of the large packets. The interleaving feature also provides a special transmit queue for the smaller, delay-sensitive packets, enabling them to be sent earlier than other flows.

Weighted fair queueing on MLP works on the packet level, not at the level of multilink fragments. Thus, if a small real-time packet gets queued behind a larger best-effort packet and no special queue has been reserved for real-time packets, the small packet will be scheduled for transmission only after all the fragments of the larger packet are scheduled for transmission.

Weighted fair queueing is now supported on all interfaces that support Multilink PPP, including MLP virtual access interfaces and virtual interface templates. Weighted fair-queueing is enabled by default.

Fair queueing on MLP overcomes a prior restriction. Previously, fair queueing was not allowed on virtual access interfaces and virtual interface templates. Interleaving provides the delay bounds for delay-sensitive voice packets on a slow link that is used for other best-effort traffic.

MLP Restrictions

Interleaving applies only to interfaces that can configure a multilink bundle interface. These restrictions include virtual templates, dialer interfaces, and ISDN BRI or PRI interfaces.

Multilink and fair queueing are not supported when a multilink bundle is off-loaded to a different system using Multichassis Multilink PPP (MMP). Thus, interleaving is not supported in MMP networking designs.

Configuring MLP Interleaving

MLP support for interleaving can be configured on virtual-templates, dialer interfaces, and ISDN BRI or PRI interfaces. To configure interleaving, complete the following tasks:

- Configure the dialer interface, BRI interface, PRI interface, or virtual template, as defined in the relevant chapters of this manual.
- Configure MLP and interleaving on the interface or template.

Note Fair queueing, which is enabled by default, must remain enabled on the interface.

To configure MLP and interleaving on a configured and operational interface or virtual interface template, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	**ppp multilink**	Enables Multilink PPP.
Step 2	**ppp multilink interleave**	Enables real-time packet interleaving.
Step 3	**ppp multilink fragment-delay** *milliseconds*	Optionally, configures a maximum fragment delay.
Step 4	**ip rtp reserve** *lowest-udp-port range-of-ports* [*maximum-bandwidth*]	Reserves a special queue for real-time packet flows to specified destination UDP ports, allowing real-time traffic to have higher priority than other flows.
Step 5	**multilink virtual-template 1**	For virtual templates only, applies the virtual template to the multilink bundle.[1]

1. This step is not used for ISDN or dialer interfaces.

Interleaving statistics can be displayed by using the **show interfaces** command, specifying the particular interface on which interleaving is enabled. Interleaving data is displayed only if there are interleaves. For example, the following line shows interleaves:

```
Output queue: 315/64/164974/31191 (size/threshold/drops/interleaves)
```

MLP Inverse Multiplexer and Distributed MLP Configuration Task List

The distributed MLP feature combines T1/E1 lines in a VIP on a Cisco 7500 series router into a bundle that has the combined bandwidth of the multiple T1/E1 lines. This is done using a VIP MLP link. You choose the number of bundles and the number of T1/E1 lines in each bundle, which allows you to increase the bandwidth of your network links beyond that of a single T1/E1 line without having to purchase a T3 line.

Non-distributed MLP can only perform limited links, with CPU usage quickly reaching 90% with only a few T1/E1 lines running MLP. With distributed MLP, you can increase the router's total capacity.

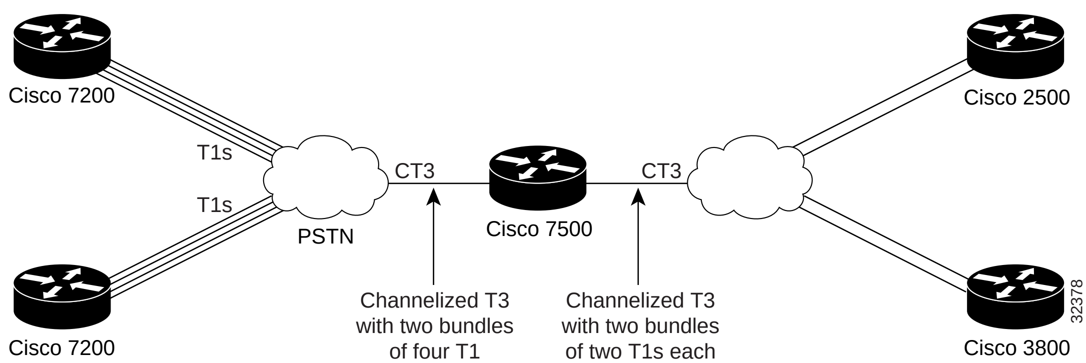
The MLP Inverse Multiplexer feature was designed for Internet service providers (ISPs) that want to have the bandwidth of multiple T1 lines with performance comparable to that of an inverse multiplexer without the need of buying standalone inverse-multiplexing equipment. A Cisco router supporting VIPs can bundle multiple T1 lines in a CT3 or CE3 interface. Bundling is more economical than purchasing an inverse multiplexer, and eliminates the need to configure another piece of equipment.

This feature supports the CT3 CE3 data rates without taxing the RSP and CPU by moving the data path to the VIP. This feature also allows remote sites to purchase multiple T1 lines instead of a T3 line, which is especially useful when the remote site does not need the bandwidth of an entire T3 line.

This feature allows multilink fragmentation to be disabled, so multilink packets are sent using Cisco Express Forwarding (CEF) on all platforms, if fragmentation is disabled. CEF is now supported with fragmentation enabled or disabled.

Figure 51 shows a typical network using a VIP MLP link. The Cisco 7500 series router is connected to the network with a CT3 line that has been configured with VIP MLP to carry two bundles of four T1 lines each. One of these bundles goes out to a Cisco 2500 series router and the other goes out to a Cisco 3800 series router.

Figure 51 Diagram of a Typical VIP MLP Topology



Before beginning the MLP Inverse Multiplexer configuration tasks, make note of the following prerequisites and restrictions.

Prerequisites

- Distributed CEF switching must be enabled for distributed MLP.
- One of the following port adapters is required:
  - CT3IP
  - PA-MC-T3
  - PA-MC-2T3+
  - PA-MC-E3
  - PA-MC-8T1
  - PA-MC-4T1
  - PA-MC-8E1
- All 16 E1s can be bundled from a PA-MC-E3 in a VIP4-80.

Restrictions

- The Multilink Inverse Multiplexer feature is supported only on the Cisco 7500 series routers.
- For bundles using IP, all lines in the bundle must have the same IP access list.
- Only one port adapter can be installed in a VIP.
- T1 and E1 lines cannot be mixed in a bundle.
- T1 lines in a bundle must have the same bandwidth.
- All lines in a bundle must have identical configurations.
- T1 lines can be combined in one bundle or up to 16 bundles per VIP.
- E1 lines can be combined in one bundle or up to 12 bundles per VIP.
- A maximum of eight T1 lines can be bundled on the VIP2-50 with two MB of SRAM.
- A maximum of 16 T1 lines can be bundled on the VIP2-50 with four or eight MB of SRAM.
- A maximum of 12 E1 lines can be bundled on the VIP2-50 with four or eight MB of SRAM.
- A maximum of 40 T1 lines can be bundled on the VIP4-80.
- Hardware compression is not supported.
- Encryption is not supported.
- Fancy/custom queueing is supported.
- MLP fragmentation is supported.
- Software compression is not recommended because CPU usage would negate performance gains.
- The maximum differential delay supported is 50 milliseconds.
- VIP CEF is limited to IP only; all other protocols are sent to the RSP.

Enabling fragmentation reduces the delay latency among bundle links, but adds some load to the CPU. Disabling fragmentation may result in better throughput.

If your data traffic is consistently of a similar size, we recommend disabling fragmentation. In this case, the benefits of fragmentation may be outweighed by the added load on the CPU.

To configure a multilink bundle, perform the tasks in the following sections:

- Enabling Distributed CEF Switching (Required for Distributed MLP)
- Creating a Multilink Bundle (Required)
- Assigning an Interface to a Multilink Bundle (Required)
- Disabling PPP Multilink Fragmentation (Optional)
- Verifying the MLP Inverse Multiplexer Configuration (Optional)

Enabling Distributed CEF Switching

To enable distributed MLP, first enable distributed CEF switching (dCEF) by using the following command in global configuration mode:

Command	Purpose
`ip cef distributed`	Enables distributed CEF switching.

Creating a Multilink Bundle

To create a multilink bundle, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**interface multilink** *group-number*	Enters multilink interface configuration mode.
Step 2	**ip address** *address mask*	Assigns an IP address to the multilink interface.
Step 3	**encapsulation ppp**	Enables PPP encapsulation.
Step 4	**ppp multilink**	Enables Multilink PPP.

Assigning an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	**no ip address**	Removes any specified IP address.
Step 2	**keepalive**	Sets the frequency of keepalive packets.
Step 3	**encapsulation ppp**	Enables PPP encapsulation.
Step 4	**multilink-group** *group-number*	Assigns the interface to a multilink bundle.
Step 5	**ppp multilink**	Enables Multilink PPP.
Step 6	**ppp authentication chap**	(Optional) Enables CHAP authentication.
Step 7	**pulse-time** *seconds*	(Optional) Configures DTR signal pulsing.

Disabling PPP Multilink Fragmentation

By default, PPP multilink fragmentation is enabled. To disable PPP multilink fragmentation, use the following command in interface configuration mode:

Command	Purpose
no ppp multilink fragmentation	(Optional) Disables PPP multilink fragmentation.

Verifying the MLP Inverse Multiplexer Configuration

To display information about the newly created multilink bundle, use the **show ppp multilink** command in EXEC mode:

```
Router# show ppp multilink
Multilink1, bundle name is group1
Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links:4 active, 0 inactive (max not set, min not set)
 Serial1/0/0:1
 Serial1/0/0/:2
 Serial1/0/0/:3
 Serial1/0/0/:4
```

Monitoring and Maintaining PPP and MLP Interfaces

To monitor and maintain virtual interfaces, use the following command in EXEC mode:

Command	Purpose
show ppp multilink	Displays MLP and MMP bundle information.

PPP and MLP Configuration Examples

Examples in the following sections section show various PPP configurations:

- CHAP with an Encrypted Password
- User Maximum Links Configuration
- MPPC Interface Configuration
- PPP Reliable Link
- MLP Examples
- MLP Interleaving and Queueing for Real-Time Traffic
- T3 Controller Configuration for an MLP Multilink Inverse Multiplexer
- Multilink Interface Configuration for Distributed MLP

CHAP with an Encrypted Password

The following examples show how to enable CHAP on serial interface 0 of three devices:

Configuration of Router yyy

```
hostname yyy
interface serial 0
 encapsulation ppp
 ppp authentication chap
username xxx password secretxy
username zzz password secretzy
```

Configuration of Router xxx

```
hostname xxx
interface serial 0
 encapsulation ppp
 ppp authentication chap

username yyy password secretxy
username zzz password secretxz
```

Configuration of Router zzz

```
hostname zzz
interface serial 0
encapsulation ppp
ppp authentication chap
username xxx password secretxz
username yyy password secretzy
```

When you look at the configuration file, the passwords will be encrypted and the display will look similar to the following:

```
hostname xxx
interface serial 0
 encapsulation ppp
 ppp authentication chap
username yyy password 7 121F0A18
username zzz password 7 1329A055
```

User Maximum Links Configuration

The following example shows how to configure the username sTephen and establish a maximum of five connections. sTephen can connect through serial interface 1/0, which has a dialer map configured for it, or through PRI interface 0/0:23, which has dialer profile interface 0 dedicated to it.

The **aaa authorization network default local** command must be configured. PPP encapsulation and authentication must be enabled on all the interfaces sTephen can connect to.

```
aaa new-model
aaa authorization network default local
enable secret saintstephen
enable password witharose
!
username sTephen user-maxlinks 5 password gardenhegoes
!
interface Serial0/0:23
 no ip address
 encapsulation ppp
 dialer pool-member 1
 ppp authentication chap
 ppp multilink
!
interface Serial1/0
 ip address 2.2.2.4 255.255.255.0
 encapsulation ppp
 dialer in-band
 dialer map ip 2.2.2.13 name sTephen 12345
 dialer-group 1
 ppp authentication chap
!
```

```
interface Dialer0
 ip address 1.1.1.4 255.255.255.0
 encapsulation ppp
 dialer remote-name sTephen
 dialer string 23456
 dialer pool 1
 dialer-group 1
 ppp authentication chap
 ppp multilink
!
dialer-list 1 protocol ip permit
```

MPPC Interface Configuration

The following example configures asynchronous interface 1 to implement MPPC and ignore the protocol field compression flag negotiated by LCP:

```
interface async1
 ip unnumbered ethernet0
 encapsulation ppp
 async default routing
 async dynamic routing
 async mode interactive
 peer default ip address 172.21.71.74
 compress mppc ignore-pfc
```

The following example creates a virtual access interface (virtual-template interface 1) and serial interface 0, which is configured for X.25 encapsulation. MPPC values are configured on the virtual-template interface and will ignore the negotiated protocol field compression flag.

```
interface ethernet0
 ip address 172.20.30.102 255.255.255.0
!
interface virtual-template1
 ip unnumbered ethernet0
 peer default ip address pool vtemp1
 compress mppc ignore-pfc
!
interface serial0
 no ipaddress
no ip mroute-cache
encapsulation x25
x25 win 7
x25 winout 7
x25 ips 512
x25 ops 512
clock rate 50000
!
ip local pool vtemp1 172.20.30.103 172.20.30.104
ip route 0.0.0.0 0.0.0.0 172.20.30.1
!
translate x25 31320000000000 virtual-template 1
```

PPP Reliable Link

The following example enables PPP reliable link and STAC compression on BRI 0:

```
interface BRI0
 description Enables stac compression on BRI 0
 ip address 172.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 172.1.1.2 name baseball 14195386368
 compress stac
 ppp authentication chap
 dialer-group 1
 ppp reliable-link
```

The following example shows output of the **show interfaces** command when PPP reliable link is enabled. The LAPB output lines indicate that PPP reliable link is provided over LAPB.

```
Router# show interface serial 0
Serial0 is up, line protocol is up
  Hardware is HD64570
  Description: connects to enkidu s 0
  Internet address is 172.21.10.10/8
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set
  LCP Open
  Open: IPCP, CDP
  LAPB DTE, state CONNECT, modulo 8, k 7, N1 12048, N2 20
      T1 3000, T2 0, interface outage (partial T3) 0, T4 0, PPP over LAPB
      VS 1, VR 1, tx NR 1, Remote VR 1, Retransmissions 0
      Queues: U/S frames 0, I frames 0, unack. 0, reTx 0
      IFRAMEs 1017/1017 RNRs 0/0 REJs 0/0 SABM/Es 1/1 FRMRs 0/0 DISCs 0/0
  Last input 00:00:18, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/64/0 (size/threshold/drops)
     Conversations  0/1 (active/max active)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 3000 bits/sec, 4 packets/sec
  5 minute output rate 3000 bits/sec, 7 packets/sec
     1365 packets input, 107665 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     2064 packets output, 109207 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 output buffer failures, 0 output buffers swapped out
     4 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

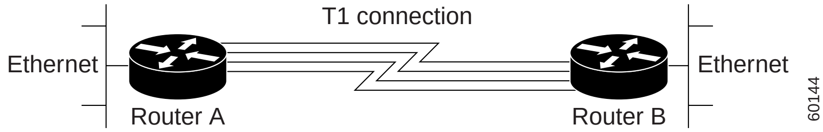
MLP Examples

The following examples configure Multilink PPP on synchronous serial interfaces, on one BRI interface, and on multiple BRIs belonging to the same dialer rotary group that is configured for Multilink PPP.

Multilink PPP on Synchronous Serial Interfaces

Multilink PPP provides characteristics most similar to hardware inverse multiplexers, with good manageability and Layer 3 services support. Figure 52 shows a typical inverse multiplexing application using two Cisco routers and Multilink PPP over four T1 lines.

Figure 52 Inverse Multiplexing Application Using Multilink PPP



The following example shows the configuration commands used to create the inverse multiplexing application:

Router A Configuration

```
hostname RouterA
!
!
username RouterB password your_password
ip subnet-zero
multilink virtual-template 1
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 ppp authentication chap
 ppp multilink
!
interface Serial0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial1
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial2
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial3
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Ethernet0
 ip address 10.17.1.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

Router B Configuration

```
hostname RouterB
!
!
username RouterB password your_password
ip subnet-zero
multilink virtual-template 1
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 ppp authentication chap
 ppp multilink
!
interface Serial0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial1
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial2
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial3
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Ethernet0
 ip address 10.17.2.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

Multilink PPP with Dialer Profiles Example

The following example indicates how to configure Multilink PPP with dialer profiles: both the physical and logical interfaces must have the **ppp multilink** command applied to it.

```
!
 interface BRI2/1
  no ip address
  encapsulation ppp
  dialer pool-member 2
  no fair-queue
  no cdp enable
  ppp authentication chap
  ppp multilink
 !
 !
 interface Dialer2
  ip address 10.1.1.1 255.255.255.0
  encapsulation ppp
  dialer remote-name bavik
  dialer string 229
  dialer pool 2
  dialer-group 1
  no fair-queue
  ppp authentication chap
  ppp multilink
 !
```

MLP on One ISDN BRI Interface

The following example enables MLP on the BRI interface 0. Because an ISDN interface is a rotary group by default, when one BRI is configured, no dialer rotary group configuration is required.

```
interface bri 0
 description connected to ntt 81012345678902
 ip address 171.1.1.7 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 30
 dialer load-threshold 40 either
 dialer map ip 171.1.1.8 name atlanta 81012345678901
 dialer-group 1
 ppp authentication pap
 ppp multilink
```

MLP on Multiple ISDN IBRI Interfaces

The following example configures multiple ISDN BRI interfaces to belong to the same dialer rotary group for Multilink PPP. The **dialer rotary-group** command is used to assign each of the ISDN BRI interfaces to that dialer rotary group.

```
interface BRI0
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface BRI1
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface BRI2
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface Dialer0
 ip address 99.0.0.2 255.0.0.0
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 500
 dialer map ip 99.0.0.1 name atlanta broadcast 81012345678901
 dialer load-threshold 30 either
 dialer-group 1
 ppp authentication chap
 ppp multilink
```

MLP Interleaving and Queueing for Real-Time Traffic

The following example defines a virtual interface template that enables MLP interleaving and a maximum real-time traffic delay of 20 milliseconds, then applies that virtual template to the MLP bundle:

```
interface virtual-template 1
 ip unnumbered ethernet 0
 ppp multilink
 ppp multilink interleave
 ppp multilink fragment-delay 20
 ip rtp interleave 32768 20 1000
multilink virtual-template 1
```

The following example enables MLP interleaving on a dialer interface that controls a rotary group of BRI interfaces. This configuration permits IP packets to trigger calls.

```
interface BRI 0
 description connected into a rotary group
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 1
 no ip address
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 2
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 3
 no ip address
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 4
 encapsulation ppp
 dialer rotary-group 1
!
interface Dialer 0
 description Dialer group controlling the BRIs
 ip address 8.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 8.1.1.2 name angus 14802616900
 dialer-group 1
 ppp authentication chap
! Enables Multilink PPP interleaving on the dialer interface and reserves
! a special queue.
 ppp multilink
 ppp multilink interleave
 ip rtp reserve 32768 20 1000
! Keeps fragments of large packets small enough to ensure delay of 20 ms or less.
 ppp multilink fragment-delay 20
dialer-list 1 protocol ip permit
```

T3 Controller Configuration for an MLP Multilink Inverse Multiplexer

In the following example, the T3 controller is configured and four channelized interfaces are created:

```
controller T3 1/0/0
framing m23
cablelength 10
t1 1 timeslots 1-24
t1 2 timeslots 1-24
t1 3 timeslots 1-24
t1 4 timeslots 1-24
```

Multilink Interface Configuration for Distributed MLP

In the following example, four multilink interfaces are created with distributed CEF switching and MLP enabled. Each of the newly created interfaces are added to a multilink bundle:

```
interface multilink1
 ip address 10.0.0.0 10.255.255.255
 ppp chap hosstname group 1
 pp multilink
 multilink-group 1

interface serial 1/0/0/:1
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp multilink
 multilink-group 1

interface serial 1/0/0/:2
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 multilink-group 1

interface serial 1/0/0/:3
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 multilink-group 1

interface serial 1/0/0/:4
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 multilink-group 1
```